Net²: A Graph Attention Network Method Customized for Pre-Placement Net Length Estimation

Zhiyao Xie[1], Rongjian Liang[2], Xiaoqing Xu[3],
Jiang Hu[2], Yixiao Duan[1], Yiran Chen[1]

Duke University[1], Texas A&M University[2], Arm Inc.[3]

Duke

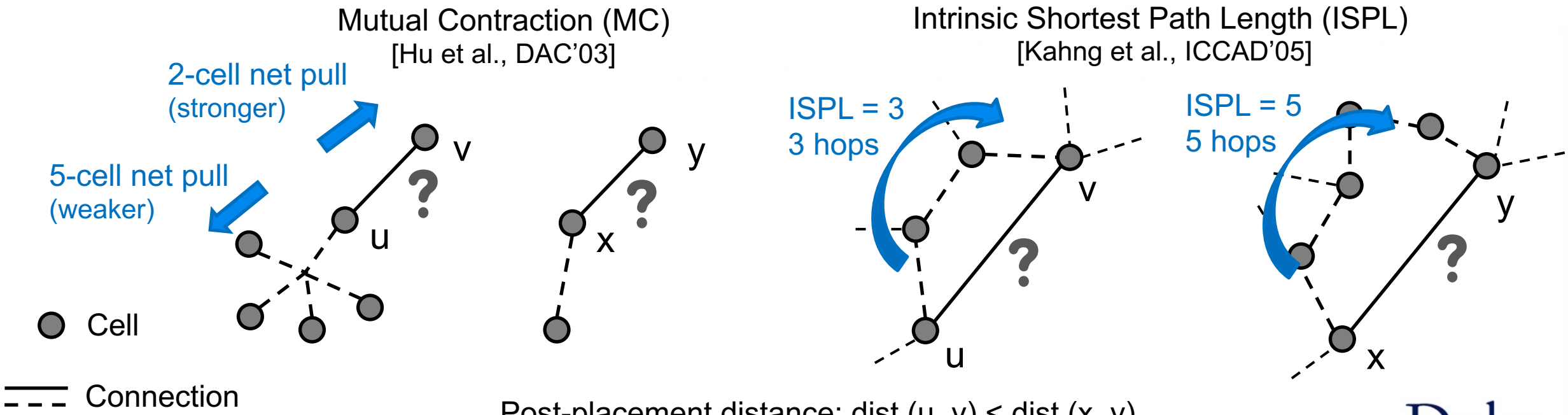
Background: Early Net Size Prediction

- Net size prediction is desired at early stage
  - Interconnect is a dominating factor for power & performance
  - Not explicitly quantified and optimized until the layout stage
- Trend in EDA industry: improve predictability at early stage
  - Physical-aware synthesis
  - Use consistent EDA engines at both logic synthesis & PnR
  - Cadence: iSpatial flow in Genus + Innovus
  - Synopsys: replace DC + ICC2 with Fusion Compiler
- Problem formulation of net size prediction
  - **Before placement, given a netlist, predict the post-placement net size (HPWL of the bounding box) of each individual net**

Duke

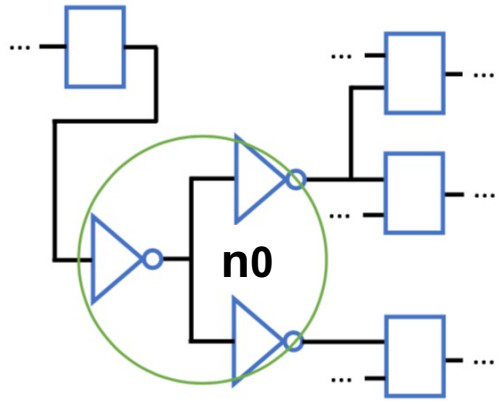
Previous Works: Early Net Size Prediction

- Previous publications on net size prediction
  - Heuristic: mutual contraction (**MC**), intrinsic shortest path length (**ISPL**)
  - Data-driven: polynomial models with local net features (**Poly**)
  - **Limitations: lack global information of the whole netlist**



An Example: Importance of Global Information

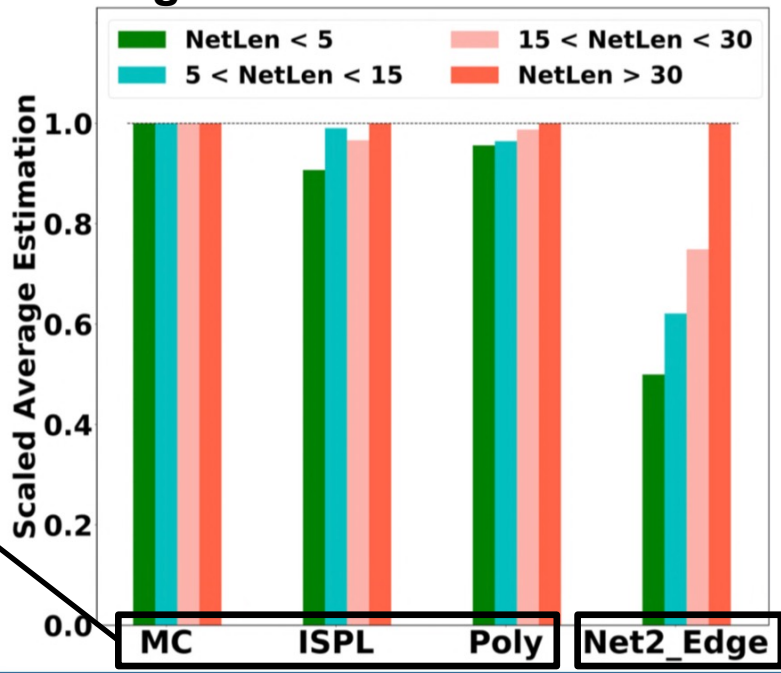
- A net **n0** with one 2-pin fan-in, one 2-pin fan-out and one 3-pin fan-out.
- In one netlist, we find 725 nets with exactly the same local info as **n0**
- The 725 'similar' nets divided into **four types** based on **actual net length:**
  - Net Length (*μm*):



Local information of net **n0** includes:

- Three basic inverters with size 1
- 2-pin fan-in, 2-pin fan-out, 3-pin fan-out

Average estimation value for each type



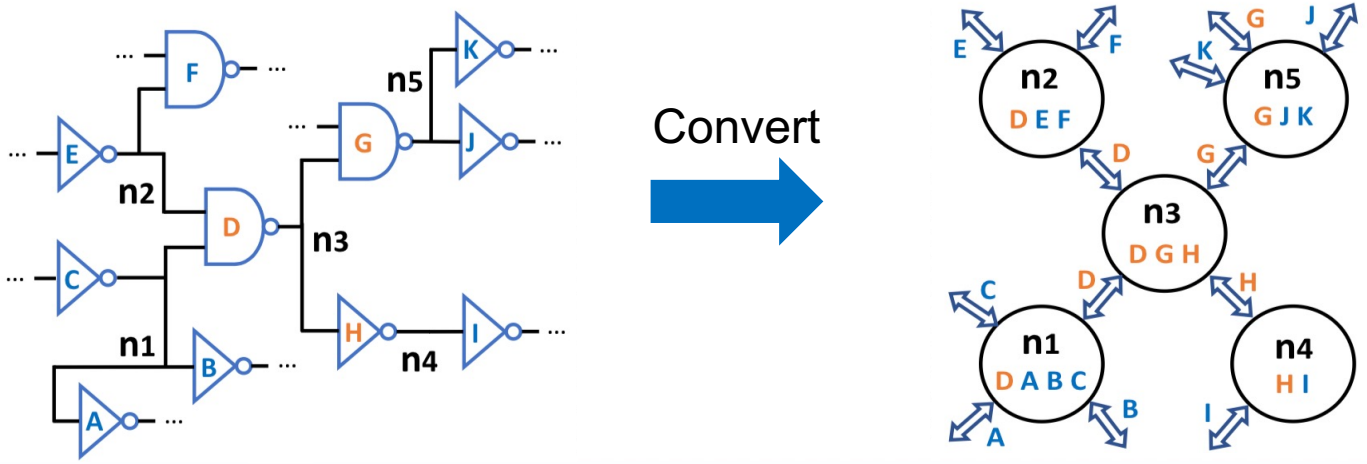
Previous methods without much global information

Part of our Net²

Duke

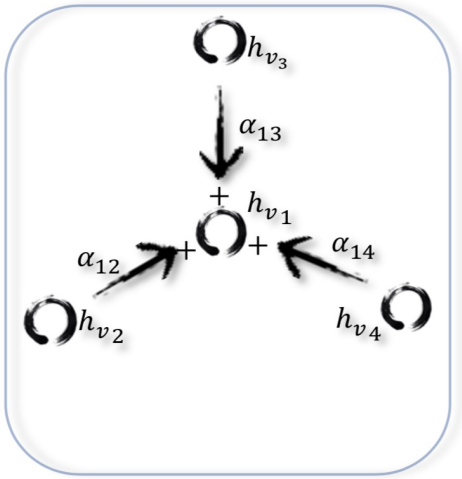
Method: Netlist as a Graph

- Convert each netlist to an input graph:
  - Each net is viewed as a node (n1, n2, n3, n4, n5)
  - The label of each node is the bounding box HPWL of the net/node
- For each net/node, its node features include:
  - #fan-in, #fan-out, driver's area, sum of area of all cells in the net,
  - The summation, mean of ([# fan-in of all fan-ins], [# fan-out of all fan-ins], [# fan-in of all fan-outs], [# fan-out of all fan-outs])



Method: the Fast Version of Net^2 → Net^{2f}

- Graph Neural Network (GNN) models
  - They are comprised of multiple sequential convolutional layers
  - Each layer generates a new embedding for each node, the input is the previous embedding of the node and its neighbors
- Representative GNN models:
  - Graph convolutional network (GCN) [Kipf et al., ICLR'16]
  - Graph sage model (Gsage) [Hamilton et al, NeurIPS'17]
  - Graph attention network (GAT) [Veličković, ICLR'17]
- Build the **fast version** of our Net^2 model, named Net^{2f}:
  - The model is built on 3 convolutional layers of GAT.
  - Customization:
    - The final embedding includes outputs of all previous layers, instead of only the last layer.
    - Combine shallow & deep feature maps. Proved useful in CNN. Not widely seen in GNN.



GNN convolutional layer*

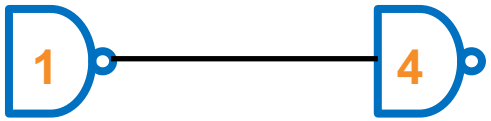
* Image from: Wu, et al., A Comprehensive Survey on Graph Neural Networks, 2019

Method: Capture the Global Information

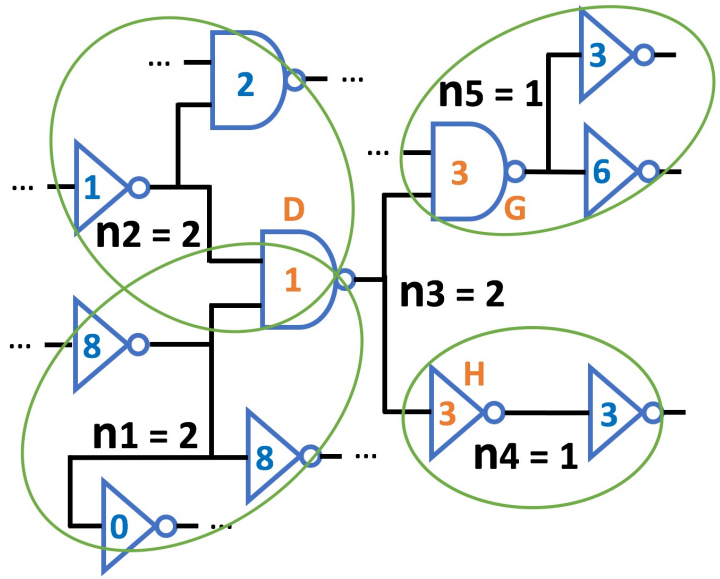
- To build the accurate version Net^{2a}:
  - Capture more global information (the topology of the whole netlist)
  - Capture the information by fast clustering (partitioning) on each netlist using h-metis
  - 1. clustering when nets viewed as node; 2. clustering when cells viewed as node
  - **Different cluster IDs indicates larger distance after placement**

Numbers on cell are cluster IDs:

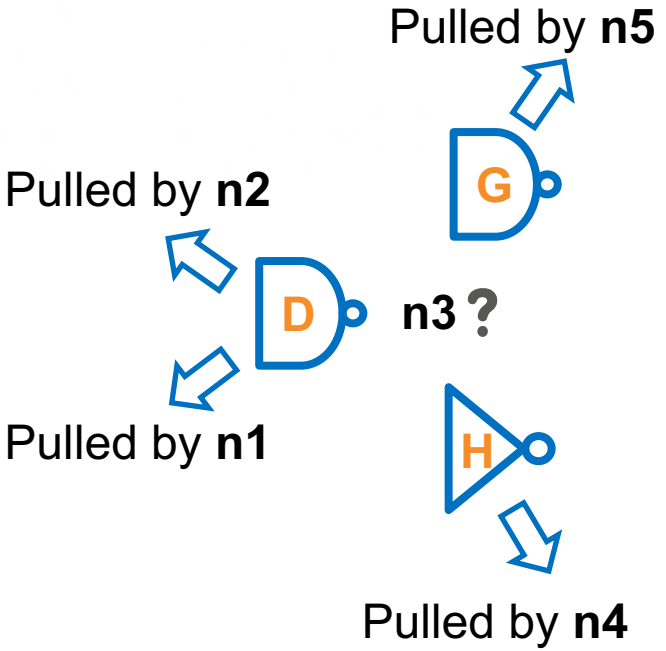
The same cluster IDs (1 == 1).
It indicates shorter distance.



The different cluster IDs (1 != 4).
It indicates longer distance.



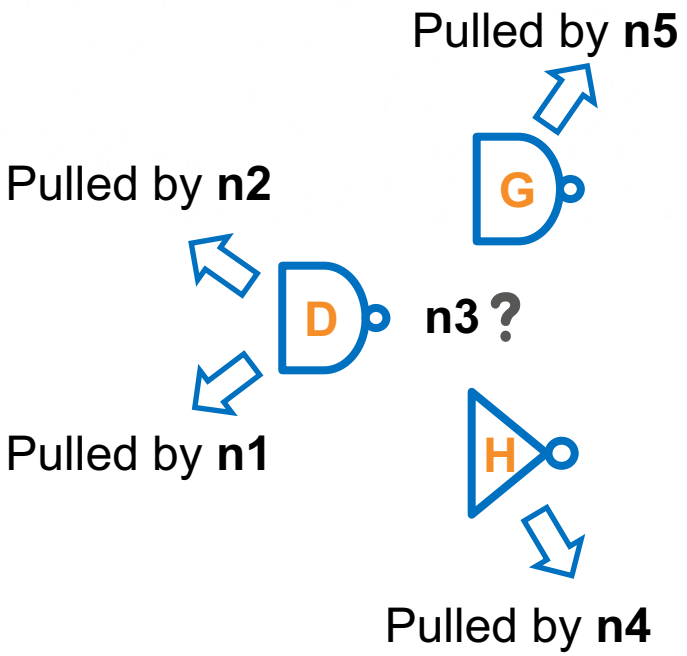
Example: study the size of **n3**.
Numbers on cell & net are cluster IDs.

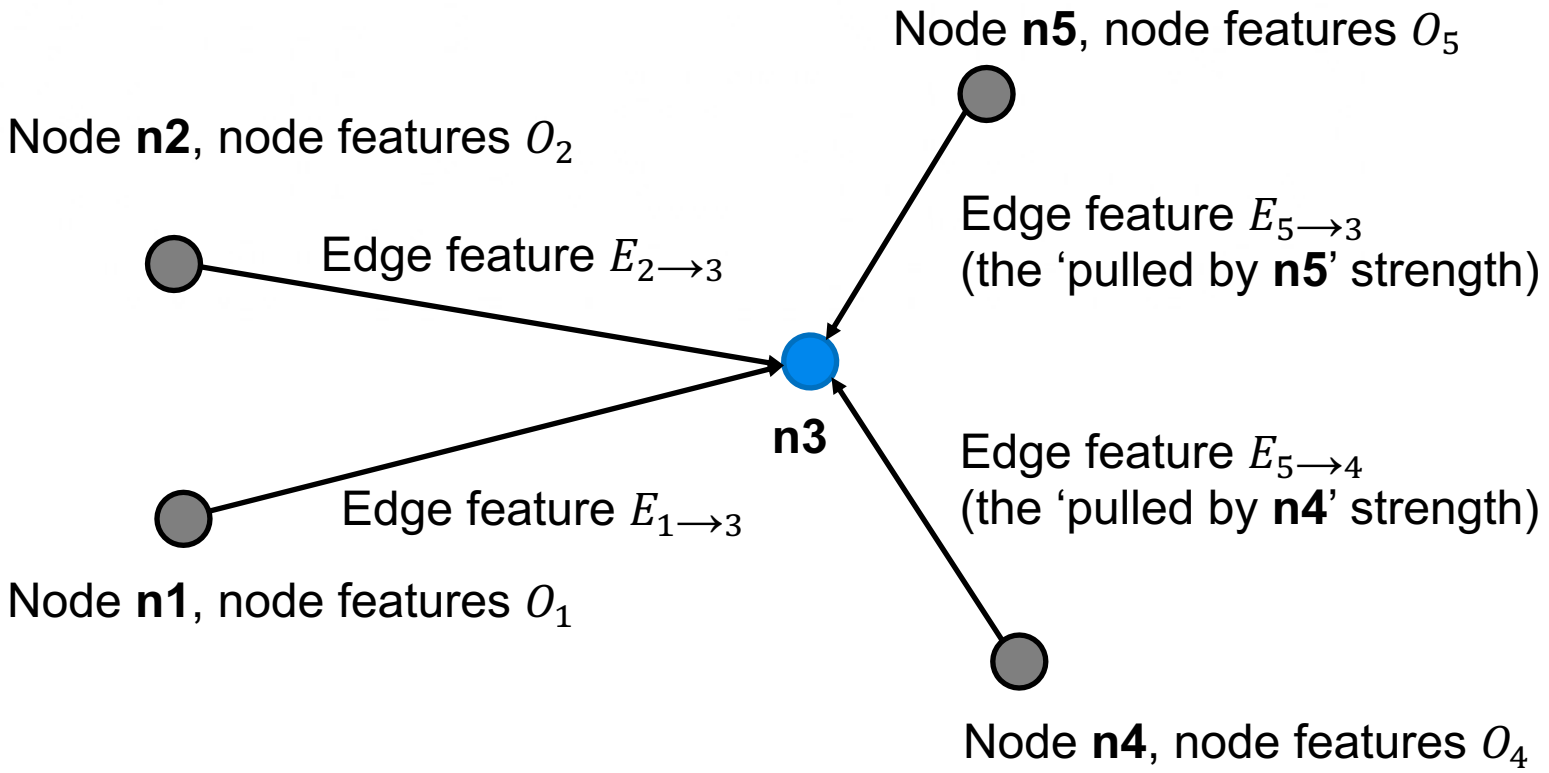


Duke

Method: Incorporate the Global Information in Edge

- Define edge features based on the captured global information
  - The directional edge feature measures the 'pulling strength' indicated by clusters IDs





Duke

Method: the Accurate Version of Net^2 → Net^{2a}

- Build the **accurate version** of our Net^2 model, named Net^{2a}:
  - Include three node convolution layers of GAT, same as the fast version Net^{2f}
  - Include additional **edge convolutional layer**, handling the edge features
  - The final embedding includes outputs of all previous layers.
- Encode the 'pulling strength' from cluster IDs as edge features:
- For node n_k, the edge convolution layer:

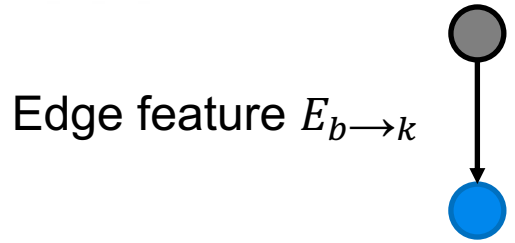
$$e_{k\_sum} = \sum_{n_b \in \mathcal{N}(n_k)} W_2 W_1 [O_k || E_{b\to k} || O_b]$$

Two-layer ANN

Go through all neighbors of n_k

Concatenate both node features and edge feature

Neighbor: Node n_b with node feature O_b



Node n_k with node feature O_k

Duke

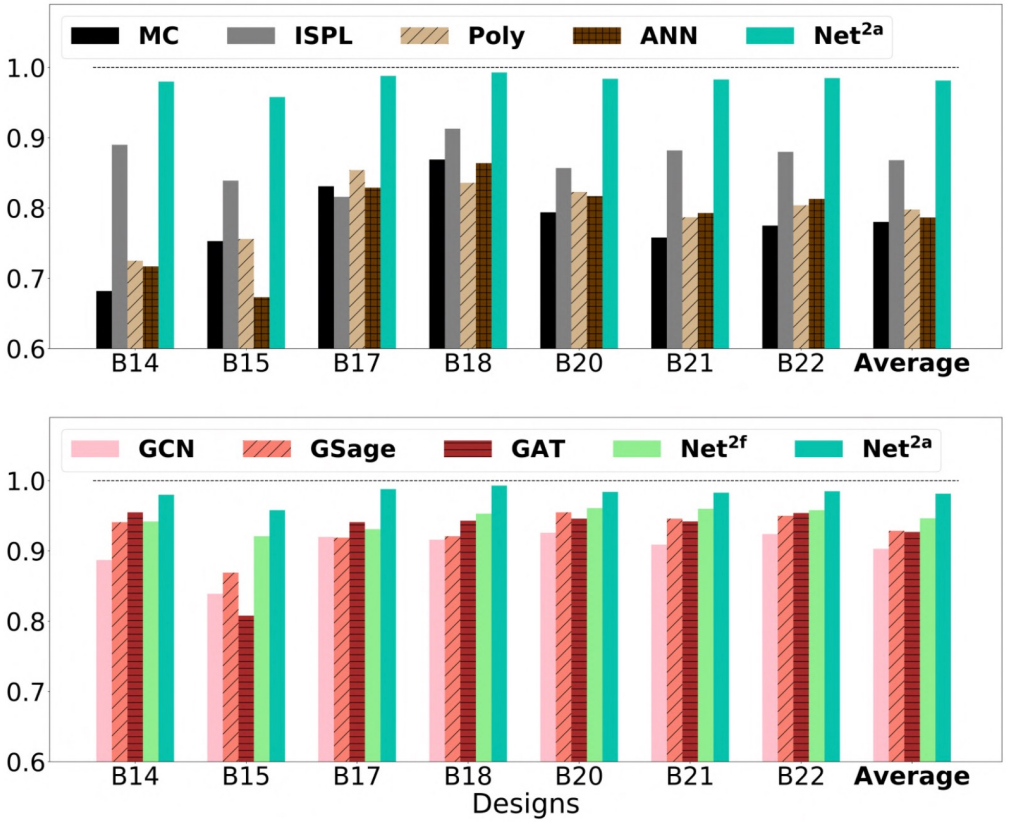
Experiment Setup

- Dataset:
  - 7 different designs from ITC'99
  - For each design, synthesize 10 different netlists with different synthesis parameters
  - Use Design Compiler (DC) and Innovus.

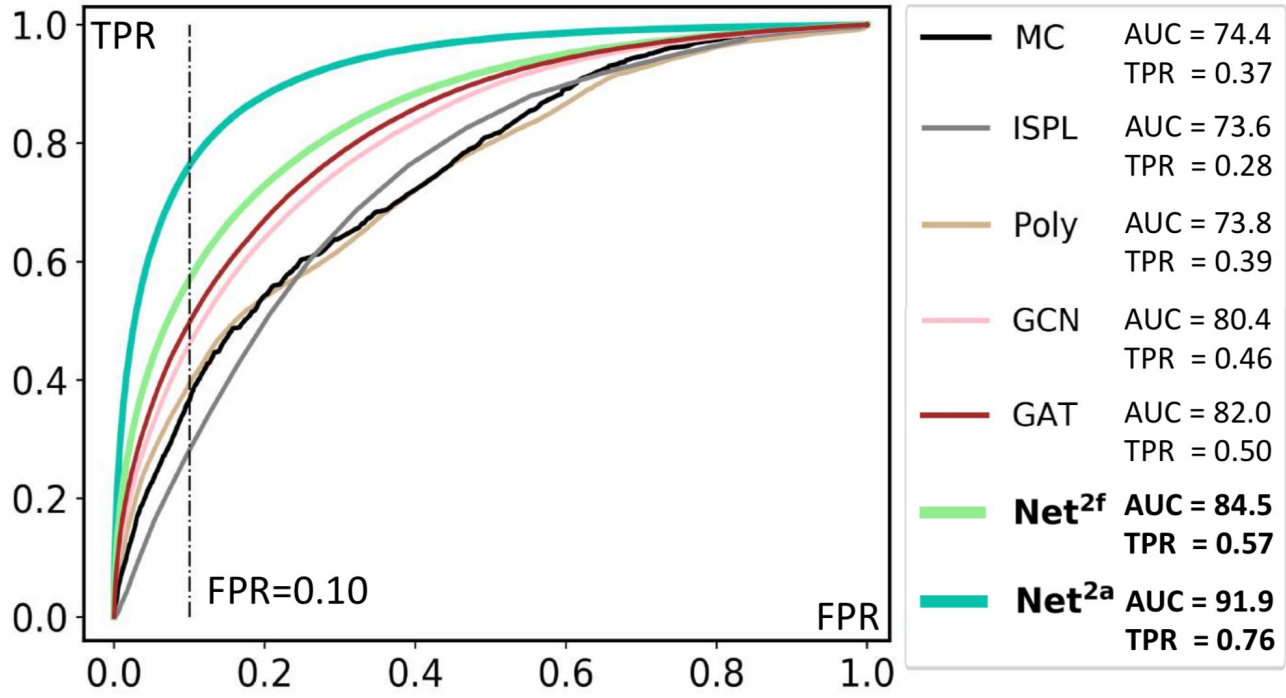
	B14	B15	B17	B18	B20	B21	B22
Smallest	13 K	5.3 K	18 K	54 K	26 K	26 K	39 K
Largest	34 K	15 K	49 K	138 K	67 K	66 K	99 K

- Train & test:
  - Net² is cross-design model
  - For each design, the model is only trained on the 60 netlists from the other six designs

Result: Accuracy for Net Size Prediction



- Divide all nets in a netlist into 20 bins by net size.
- Measure averaged predicted net size at each bin.
- Report the **correlation R** between prediction and label in these 20 bins.



- **ROC AUC** in identifying the 10% longest nets.

Trend: Net2a > Net2f > GAT > other methods.

Duke

Result: Accuracy for Path Length Prediction

- Path length prediction:
  - The path length correlates with post-placement wire delay on the path
  - The path length is the summation of the net sizes over all nets on this path
  - Only select timing-critical paths according to the pre-placement timing report

Methods	B14	B15	B17	B18	B20	B21	B22	Ave
ISPL	58.9	57.5	56.5	74.0	72.5	63.0	75.5	65.4
Poly	65.5	80.0	78.0	68.0	82.0	85.0	84.0	77.5
ANN	68.0	76.0	80.0	69.0	78.5	82.0	75.5	75.6
GCN	63.5	75.0	86.5	56.0	82.0	81.5	85.5	75.7
GSage	65.0	88.0	93.0	77.0	81.5	67.0	80.0	78.8
GAT	63.0	92.0	95.0	83.5	83.5	76.0	89.5	83.2
Net^{2f}	79.0	88.5	97.5	84.0	75.5	83.0	92.0	85.6
Net^{2a}	86.5	95.0	96.0	90.5	90.5	93.5	95.5	92.5

Identify the 10% longest paths in ROC AUC (%)

Methods	B14	B15	B17	B18	B20	B21	B22	Ave
ISPL	67.1	55.0	58.2	77.4	68.9	59.7	69.5	65.1
Poly	83.9	86.6	83.3	70.4	83.4	80.4	86.3	82.0
ANN	82.0	74.8	75.3	68.1	81.9	65.4	80.5	75.4
GCN	74.5	85.9	83.0	62.4	83.4	81.0	86.2	79.5
GSage	84.2	92.5	83.9	75.3	89.1	62.8	88.1	82.3
GAT	82.4	93.5	85.1	80.6	89.7	87.5	88.2	86.7
Net^{2f}	87.3	92.7	87.6	93.1	91.1	91.2	86.9	90.0
Net^{2a}	96.8	97.0	91.4	95.9	92.2	94.2	94.4	94.6

Comparing pair of paths by lengths.
Percentage of correct predictions (%).

Trend: Net^{2a} > Net^{2f} > GAT > other methods.

Duke

Result: Inference Speed Comparison

- Inference speed comparison:
  - The runtime is averaged over all netlists
  - The '**place**' time includes the runtime of placement algorithm only
  - Clustering / partitioning takes most of the runtime of Net^{2a}
  - Fast version Net^{2f} is >1000× faster than placement
  - Accurate version Net^{2a} is >10× faster than placement

Inference speed comparison (in seconds)

Design	Place	Partition	Net^{2f} Infer	Net^{2a} Infer	Net^{2f} Speedup	Net^{2a} Speedup
Ave	97.8	7.0	0.05	0.07	1.7K ×	14.3×

Duke

Conclusion

- We propose a graph attention network (GAT)-based method, which is customized for individual net length estimation
- We capture global information by clustering / partitioning, and incorporate the captured information into edge features
- We propose a fast version Net^{2f}, which is 1000 $\times$ faster than placement
- We propose an accuracy-centric version Net^{2a}, which efficiently utilizes global information and achieves significantly better accuracy than previous works.

Duke

Thank You!